

141

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

Artificial Intelligence
Memo. No. 164.

September 1968

Producing Memos. using TJ6, TECO
and the type 37 teletype.

Larry Kraskauer

Introduction

This memo describes the TJ6 type justifying program, which can be used in the production of memos, such as this one. In addition, sections III and IV of this memo contain related information about TECO and the type 37 teletype, thus gathering most of the information needed for producing writeups into one location. A sample of input to TJ6 is given in section V, and is in fact the very input used to produce this page of output.

The output from TJ6 may be either justified text, with the right margin exactly aligned, as in this introduction, or it may be "filled" text, with the right margin only approximately aligned. Since I do not personally like the appearance of justified text, the remainder of this memo will not be justified, but this decision, of course, rests with each particular user.

The sections of this report are:

- I. Introduction
- II. Using TJ6
- III. Inserting lower case letters into the TECO buffer
- IV. How to use a type 37 teletype
- V. Sample TJ6 input

Section II:

Using TJ6

TJ6 was written by Richard Greenblatt to facilitate the writing of his paper on the chess program. It was not originally intended to be a system program for general use, and perhaps as a result of this, it can sometimes be rather difficult to use. It tends to be rather finicky about receiving its commands in exactly the right format, for example, and there are a number of lingering bugs which pop up unpredictably now and then, which has driven more than one user up a wall. It has recently been used in the production of a few memos, however, and many of the worst bugs have been removed.

TJ6 generally takes its input from a file device (UTn or DDn). It is line oriented. A line of input is either a command line or a text line. Blank lines are completely ignored, and in fact, to get around various bugs in TJ6, it generally pays to both precede and follow command lines with blank lines. A line is a command line if the first character in the line is a ".", with one exception, which will be noted later. Some commands are followed by a numeric (decimal integer) argument, which will be denoted below by <arg>. Most commands have a long and a short form. In use, the command must be preceded by a ".", which must be the first character on the command line. The commands are:

<long form>
<short form>

<Explanation>

end
(no short form)

Must be used to mark the end of the input text.

indent
in

Causes the left margin to be indented <arg> spaces from the physical left margin. The indentation will continue until altered by another command. The command ".indent 0" causes indenting to be stopped.

rindent
rin

Causes the left margin to be indented <arg> spaces from where the left margin was before the command. <arg> may be negative.

undent
un

This command has effect for only the next line of output text, causing it to be started <arg> spaces to the left of the current left margin. Undent commands are given within the scope of indent commands, to temporarily suspend the indenting for one line.

space
sp

In the output text, insert <arg> blank lines.

page
pa

In the output text, skip to the top of a new page.

center
ce

Center the next line in the input text.

litera
lit

Take the next line literally, disregarding a . at the start. This is the only way to have a text line in the input text with a . at the beginning, and is the exception mentioned earlier.

comment
c

Treat the remainder of the command line as a comment.

fill
fi

This command causes the right margin of the output text to be roughly aligned, as it would be by a typist, without actually right justifying the text.

adjust
ad

This command enters a mode in which extra spaces are inserted into the output lines as needed in order to line up the right margin of the output text.

nofill
nf

Turn off the fill mode

nojust
nj

Turn off the adjust mode.

break
br

Lines of the input text will not be filled across this command.

single
ss

Cause the output text to be single spaced.

double
ds

Cause the output text to be double spaced.

header
he

The remainder of the command line after the space which follows the command itself is taken as a heading which is to be typed in the upper right hand corner of each page of the output text, followed immediately by the page number (with no intervening spaces.

line
l

Set line length to <arg> characters. TJS will insert spaces at the beginning of each line if needed to cause lines of the specified length to be centered on the paper.

paperw
(no short form)

Set the full page paper width to <arg> spaces. This command is used to inform the program of the actual width of the paper being used.

paper
pl

Set paper length to <arg> lines.

Console operation of TJ6

To use TJ6, load it and type as if to MIDAS. If you are not familiar with the MIDAS console command format, read on.

Assume you have prepared a file for processing by TJ6, and it is on device DEV, with file names fn1 and fn2. Assume further that you want your output on device ODV, filed under the names fn3 and fn4. Load TJ6 by typing "TJ6|H" (it is currently on the system device), and type:

```
ODV:fn3 fn4_DEV:fn1 fn2<carriage return>
```

This command string will cause TJ6 to type the first page of the output. Typing further carriage returns will cause additional pages to be typed out.

The character "_" on the type 37 is equivalent to <BACKARROW> on a type 35 (SHIFT O on a G.E. Console).

The output device need not be a file device, in which case the file names fn3 and fn4 may be omitted. For example, the output device may be LPT, the line printer; or it may be TTY, the console you are using. The line printer gives you a quick look at what your output will be like, but has only upper case characters. If you are using a type 37 teletype (see the next section of this writeup), you may obtain output containing both upper and lower case characters. In this case, you will want to

local line feed to a fresh section of the paper before typing the final carriage return which terminates the command string, as the output will commence as soon as that carriage return is typed.

section III:

Inserting lower case letters into the TECO buffer

Case 1: You are logged in at a type 37 teletype (see section IV of this memo). In this case, there is no problem. Simply use TECO as usual. When you type a lower case letter, it will be inserted into the buffer as a lower case letter.

Case 2: You are at some other sort of console (type 35 tty, G.E. Console). You now are in trouble, since these consoles will neither accept input nor produce output in lower case. In order to get around this problem, a special mode has been added to TECO. To enter this mode, type the command "-i<dollar sign>". Note that if you type this command by itself, it will appear on your console or paper as "-i\$\$\$". but the first "\$" is a real dollar sign, whereas the second two are really alt. modes. TECO responds to this command by retyping its title, and entering -i\$ mode. In this mode, letters which are typed are converted to lower case, unless they are preceded by a "/", in which case they are left in upper case. Thus the command string "I/MAC/HAC\$\$" causes the characters "MacHac" to be inserted into the buffer. To insert a "/", type "//".

Since your console cannot output lower case characters any more than it can accept them as input, if you cause the contents of the buffer

to be displayed, upper and lower case characters cannot be distinguished. This may possibly be changed in the near future so that in -l\$ mode, upper case letters have slashes (/) added in front of them (i.e. output in the same form as the input).

To return to the regular TECO mode, type "O\$".

Section IV

How to use a type 37 teletype

1. Press the switch labeled "DATA".
2. When you get a dial tone, dial the computer. If you are at a 10 character per second teletype, dial 229. If you are at a 15 cps tty, dial 474. One can distinguish between 10 and 15 cps type 37 teletypes by the location of the telephone dial. On a 10 cps type 37, the telephone unit is on the table alongside the teletype unit, while on a 15 cps type 37, the telephone unit is vertically mounted below the table, at about knee level. The type 37 belonging to the AI group is 10 cps. A 15 cps type 37 is available in room 931 (through door 926, just to the right of the PDF-6).
3. When the computer answers with a "beep", press the "RESTORE" button on a 10 cps teletype (on the telephone unit). At a 15 cps teletype, press the "PROCEDE" button (on the left side, just above the keyboard).
4. Type a |Z ("control Z"). Note that the character set of a type 37 tty is different from that of an ordinary (type 35) tty, and that certain characters must be typed differently as well. The two most important transformations to note: on a type 37, type <PREFIX> instead of <ALT. MODE>, and type <BACKSPACE> instead of |H ("control H").

5. Typing slowly, log in as usual with <PREFIX>U. Note that characters that you type will be echoed twice, as the type 37 is still in half-duplex mode. It must be put into full-duplex mode before proceeding. To do this:

6. Type "lock<BACKSPACE>"

7. When LOCK responds (with a "_"), type "i", then "q". This will cause LOCK to initialize the type 37 into full-duplex mode, then to kill itself.

CONGRATULATIONS! You are now ready to use your type 37 as a normal (?) ITS time sharing console!

Section V:

Sample TJS input

This sample input is in fact exactly that used to produce section I of this memo.

```
.comment tjs memo14
```

```
.header TJS page
```

```
.paperw 92
```

```
.line 74
```

```
.paper 50
```

```
.double
```

```
.fill
```

```
.adjust
```

```
.center
```

```
Introduction
```

```
.sp 1
```

This memo describes the TJS type justifying program, which can be used in the production of memos, such as this one. In addition, sections III and IV of this memo contain related information about TECO and the type 37 teletype, thus gathering most of the information needed for producing writeups into one location. A sample of input to TJS is given in section V, and is in fact the very input used to produce this page of output.

```
.sp 1
```

The output from TJS may be either justified text, with the right margin exactly aligned, as in this introduction, or it may be "filled" text, with the right margin only approximately aligned. Since I do not personally like the appearance of justified text, the remainder of this memo will not be justified, but this decision, of course, rests with each particular user.

.sp 1

The sections of this report are:

.nojust

.nofill

.single

I. Introduction

II. Using TJC

III. Inserting lower case letters into the TECO buffer

IV. How to use a type 37 teletype

V. Sample TJC input

.fill

.page

.double